

**Notes:**

- For each user story, task is listed in the order that it should logically be completed.
- At the end, there is a Task Summary that aggregates duplicate tasks among all user stories that are to be implemented and lists them in order to be logically completed.
- The unit of effort used is Pair-Programming Hours, which is defined as the number of hours a pair will each spend on to finish implementation.

**User Stories**

- Create account and subscribe to newsletter for updates:
  - **Description:** I am a user who has multiple online accounts, including accounts for banking, shopping, and other financial services. I use applications that store and protect my passwords, and I know which ones are currently the best. However, if one of applications is proven to fail in time as hackers become more sophisticated, or if there is a new application that offers even better security, I need to know immediately. I can create an account with the website and subscribe to the newsletter.
  - **Due Date:** this coming Tuesday
  - **Tasks:** Develop a homepage UI, a redirect button for the create account page, a form for personal info along with a button to submit the info, and a username/password validation for future logins. Also develop an account homepage, and a clickable option to subscribe to the website's newsletter from the account homepage.
  - **Est. Implementation Time:** 3 Pair-Programming Hours
- Rate and Review Service
  - **Description:** In service page, as a User, I can rate the service using one to five stars and leave a review for that service.
  - **Due Date:** this coming Tuesday
  - **Tasks:** Develop a clickable 1-5 star rating button(s) for each service page UI. Also add a button to each page for the user to add a review. The button should check to make sure the user has an account and is logged in. If the user is not logged in, clicking this button will instead redirect to the login page. If the user has the proper credentials, then clicking the Add Review button will open a form to allow the user to enter text and submit it, which will then add the review to the list of reviews on the service page.
  - **Est. Implementation Time:** 2 Pair-Programming Hours
- Viewing Guide In Specific Category
  - **Description:** As a User, I can view different categories of guides on the homepage with clear and good-looking icons. After I click one category, I can

view a list of guides in that category. I can click on a guide and view it. The guide will have title, author, date written, main information and links to individual services which will help me understand my needs better.

- **Due Date:** next Tuesday
  - **Tasks:** Develop a homepage UI, and a clickable button leading to the guides page UI. From here, users can see a list of different guides in a populated list, with tabs at the top for category, date uploaded, rating, etc. Clicking on category will sort the guides by content type. Clicking on a guide name will list the title, author, date written, and a text block with information about the topic. At the bottom of the guide page, a small list will be populated with the most popular services for that content type.
  - **Est. Implementation Time:** 10 Pair-Programming Hours
- Parents Protecting Their Kids
    - **Description:** As an older parent with very little internet experience, I want to learn how predators can target my children online and how to protect their privacy. From the home page, I can click on the Search Guides button to bring me to a list of guides. Since I don't know what services I may be looking for, I can go through these guides and just understand the services that are available and how to use them.
    - **Due Date:** next Tuesday
    - **Tasks:** Develop homepage UI and clickable button that redirects to the Search Guides page. This page should automatically populate with the guides in the database, sorted by rating. Each guide name in the list should be a button that allows for a redirect to the page for that specific guide. The guide pages should provide a text output for the service type.
    - **Est. Implementation Time:** 10 Pair-Programming Hours
- View top-ranked Services
    - **Description:** As a User, I can view top-ranked services on the homepage. I can also view top-ranked services for a specific technology.
    - **Due Date:** next Tuesday
    - **Tasks:** Develop homepage UI and clickable services button. Clicking the services button will present a populated list of services sorted by rating. The top of the services list will have several clickable filter options, such as category for sorting services by specific technology types.
    - **Est. Implementation Time:** 9 Pair-Programming Hours
- Manage Bookmarks
    - **Description:** As a User, I don't always have time to read articles, but I want to be able to return to them later for future reference. After logging in and finding a really interesting guide, I can click on the bookmark button to mark it. I can close this page and go back to it later when I have the time to read.

- **Due Date:** next Tuesday
- **Tasks:** Develop a user account homepage with a list of previously bookmarked guides and services based on the user's account identity. This page will have clickable delete buttons for the bookmarks, which will allow users to remove the bookmark from their user account database of bookmarked items. Develop a clickable bookmark button for each guide and service entry in the primary database that will link the viewed item to the user account identity, and copy/link the viewed item to the user's bookmark database.
- **Est. Implementation Time:** 4 Pair-Programming Hours
- Remove inappropriate reviews
  - **Description:** As an Administrator User, I need to go through reviews for services and guides, and remove any reviews not appropriate. Removed reviews are not viewable and do not affect the rating of a service or guide.
  - **Due Date:** not due/implemented (lower priority)
  - **Tasks:** Develop an account identity setting known as Administrator. In the system this should be as simple as an admin flag being turned on or off. If the user has an account flagged as Administrator, then provide them with an additional button when viewing service and guide pages that allows them to remove reviews. Removing reviews will prevent them from appearing on the service page UI, and will unlink them from the rating system.
  - **Est. Implementation Time:** 2 Pair-Programming Hours
- Block regular users from abusing their privileges
  - **Description:** As an Administrator User, I can block regular users who may be stepping out of line as per website and account usage and etiquette. This may include leaving many bad reviews, attempted hacking from same IP, spamming script to refresh a page, etc.
  - **Due Date:** not due/implemented (lower priority)
  - **Tasks:** For administrator-level accounts, username on reviews will be clickable, and will redirect to a special page for managing that user's account privileges. An administrator-level account can choose to click a button that reduces the ability of the "lower-tier" user to rate services/guides and add reviews, effectively revoking all account-level permissions of that specific user. This should be reversible, perhaps based on elapsed time.
  - **Est. Implementation Time:** 4 Pair-Programming Hours
- Editing and deleting guides for the website
  - **Description:** As a Content Author, I want to be able to update and delete my guides and service recommendations so that what I post stay relevant with current technology and services.
  - **Due Date:** not due/implemented (lower priority)

- **Tasks:** Develop an account level known as Content Author, which would have more permissions than a standard user, but fewer permissions than an Administrator-level account. Content authors should have an extra clickable when entering the Guides page that allows them to Add a Guide Entry. From the user account page, these users should have an extra clickable to Update or Delete any guides or service reviews they have created in the past. This will modify the entries in the database, or remove them entirely.
  - **Est. Implementation Time:** 9 Pair-Programming Hours
- Retrieving a lost account
  - **Description:** As a User, I can click on a Forget username and/or password link, which will take me through steps to identify myself and help me to access my account when I forget the credentials necessary to login.
  - **Due Date:** not due/implemented (lower priority)
  - **Tasks:** Develop an extra clickable in the Login screen for account recovery (two buttons, one for Forgot username and another for Forgot Password). Clicking one of these buttons will redirect to another page requesting the user to enter their account email address. After submitting this email address the system will send an email to the address with the requested username or password.
  - **Est. Implementation Time:** 3 Pair-Programming Hours
- Hosting a service
  - **Description:** As a Service Owner, I can use my unique account privilege to setup a page to advertise a security service I am providing, as well as upload the related software for purchase and download.
  - **Due Date:** not due/implemented (future feature)
  - **Tasks:** Develop an additional account privilege flag, Service Owner. Service Owners will have permissions that allow them to create new service entries from the Search Services or Account homepage. This should appear in the form of clickable buttons/links. From the user account homepage, the user will have extra clickables to update/edit or delete any service entries they have created in the past. When creating a new entry, the user will be redirected to a form that allows them to fill in information for the service they are adding, including an upload button for uploading software to the database (this may be a bit harder than just creating a form where they can add a redirect to their software homepage), as well as setup payment options
  - **Est. Implementation Time:** 24 Pair-Programming Hours
- Advance Searching Specific Content as an Internet Savvy User
  - **Description:** As an internet user with at least a basic understanding of VPNs, scriptblockers, adblockers, etc., I need access to a comprehensive list of top-rated content in privacy services I'm interested in. From the home page, I click on the Search Services option, which brings me to a list of top rated

services. I use the advance search to sort the specific category of services I'm interested in, at which point the list is now curated to services I'm interested in, and it's sorted by highest rating.

- **Due Date:** not due/implemented (lower priority)
  - **Tasks:** Develop the homepage UI and the clickable button redirects to the Services and search page. Make it so that the page can be populated by the database even if no search terms are entered. Entries in the database should be listed based on rating flags. Develop the filter option for the search engine, with the ability to sort by type of content. Also develop the page UI to allow for more advanced filter options for the search. Develop the ability to click each content entry on the list to redirect to that service's page.
  - **Est. Implementation Time:** 24 Pair-Programming Hours
- Keep Bad Scripts Out
    - **Description:** As a user that roams the internet and often runs into clickable links on forums, I want to find the best way to block harmful scripts that may compromise my system. From the home page, I click on the Search Guides button, which brings me to a list of guides. Reading some guides give me an idea of the services I should be looking for, although I don't want to limit myself to those suggested in the guides. I go to Search Services and find script and ad blockers, which provide me with a much more comprehensive list. From here, I examine each service's page, and click on links to visit external sites and purchase some.
    - **Due Date:** not due/implemented (similar due user story that covers same essential task implementation)
    - **Tasks:** Develop homepage UI with clickable button redirects to Search Guides or Search Services. Guides page should automatically populate with guides from the database, sorted by rating. Develop the ability to Go Back from this page to the home page, and after clicking buttons to reach the services page, allow users to enter text for content types. Text entry should not be case sensitive, and should autocorrect to some degree (such as if the user puts a space in scriptblocker). A search should return a list of clickable service names that redirect to service information pages.
    - **Est. Implementation Time:** 8 Pair-Programming Hours
- Search for Guides on a Specific Service
    - **Description:** I frequently browse the internet, but I am uncomfortable with the fact that my ISP knows my browsing history. I recently learned that I can evade this using a VPN, but I have no idea where to start. I want a website that gives me detailed information about the best VPN services, their pros and cons, and how to use them. I can navigate to the homepage, clicks on "search guides", filter my search by "VPN", and read the top articles on VPN services.

- **Due Date:** not due/implemented (similar due user story that covers same essential task implementation)
  - **Tasks:** Develop homepage UI, clickable search guides button, guides homepage, search bar with the ability to search for specific terms (allow for some variation in the terms, such as all lowercase), and the ability to filter the initial population of the guides page by content type. Also develop the ability to click the names of each guide to redirect to a page for each guide.
  - **Est. Implementation Time:** 6 Pair-Programming Hours
- Write Guides on a Specific Service
    - **Description:** I am an avid internet user, and I tend to have the scoop on new services. I want a place to share my knowledge where it will be immediately available to a community of users who are specifically looking for this kind of information. I can navigate to the homepage, create an account, designate myself as a tech savvy user, wait approval to be content author, and afterwards create services and guides for these services as I learn about them.
    - **Due Date:** not due/implemented (lower priority)
    - **Tasks:** Develop homepage UI, create account button, redirect to a form entry for personal information that allow a clickable box to designate tech knowledge, with tech savvy being an option (not necessarily this exact designation, but the same idea). Develop a username/password validation procedure for account login from the homepage, with account permissions allowing for new content to be added to the database. Develop request system for administrator to approve account as content author. When creating new content, content type is selected and a form is opened for the title of the content, the developer, the description, etc.
    - **Est. Implementation Time:** 13 Pair-Programming Hours
- Browse the Database for Relevant Topics
    - **Description:** I know virtually nothing about the internet. I don't know how to use Google properly to filter my results, so I need an interface that's a little more user friendly. I navigate to the homepage, clicks on "search services", and can easily filter the services I am looking for by category, without the need to target precise keywords in a search box like I would in a Google search.
    - **Due Date:** not due/implemented (lower priority)
    - **Tasks:** Develop homepage UI, clickable buttons for Search services, and a populated list/search function that allow for filtering based on content type.
    - **Est. Implementation Time:** 4 Pair-Programming Hours

**Tasks Summary:** list below indicates in order of logical completion. Pending priority, some features below may not be implemented since they are not part of high-priority user stories.

1. Develop a homepage UI with clickables for Search Guides, Search Services, Create Account, Login, Subscribe to newsletter. Each of these clickables should redirect to another page.
  - a. **Est. Implementation Time:** 3 Pair-Programming Hours
2. Develop page UI for Login page with form entry for Username and Password. This page should also have two clickables in addition to the Submit button: Forgot Username and Forgot Password. Each of these should redirect to an email entry form page. Entering proper credentials and clicking submit should redirect the user to their user account homepage.
  - a. **Est. Implementation Time:** 3 Pair-Programming Hours
3. Develop a page for Create Account. This should be primarily a form allowing for information entry, and also allowing the user to select their experience level from a list of selectable options presented as drop-down menu. Clicking submit at the bottom of the page after entering the information required, including email, should automatically email the user an account validation email, with a clickable link that takes them directly to their user account page.
  - a. **Est. Implementation Time:** 5 Pair-Programming Hours
4. Develop a user account page, which contains any bookmarked items in a list, clickable buttons for adding/editing/deleting guides/services (if the user meets the account privilege requirements), as well as redirects to the homepage so the user can Search Guides or Services.
  - a. **Est. Implementation Time:** 8 Pair-Programming Hours
5. Develop a Guides page UI, with a populated list of guides in the database, sorted by user rating, and also including filter options at the top of the list such as Date Uploaded, User Rating, Guide Type, etc. The Guides page should include a search bar that allows the user to enter text for search guidelines, and a clickable submit button that returns a list based on search terms. Search terms should not be case sensitive, and should allow some margin for error.
  - a. **Est. Implementation Time:** 7 Pair-Programming Hours
6. Develop a Services page UI, with a populated list of services in the database, sorted by user rating, and also including filter options at the top of the list such as Date Uploaded, User Rating, Service Type, etc. The Services page should include a search bar that allows the user to enter text for search guidelines, and a clickable submit button that returns a list based on search terms. Search terms should not be case sensitive, and should allow some margin for error. There should also be an Advanced Search button, that allows users to further refine their search.
  - a. **Est. Implementation Time:** 9 Pair-Programming Hours
7. Develop UI pages for each entry in the database. The guide and services pages should allow for users with even basic account privileges to rate 1-5 stars with a click, and also click an extra button to Add a Review. Each page should list relevant information for the

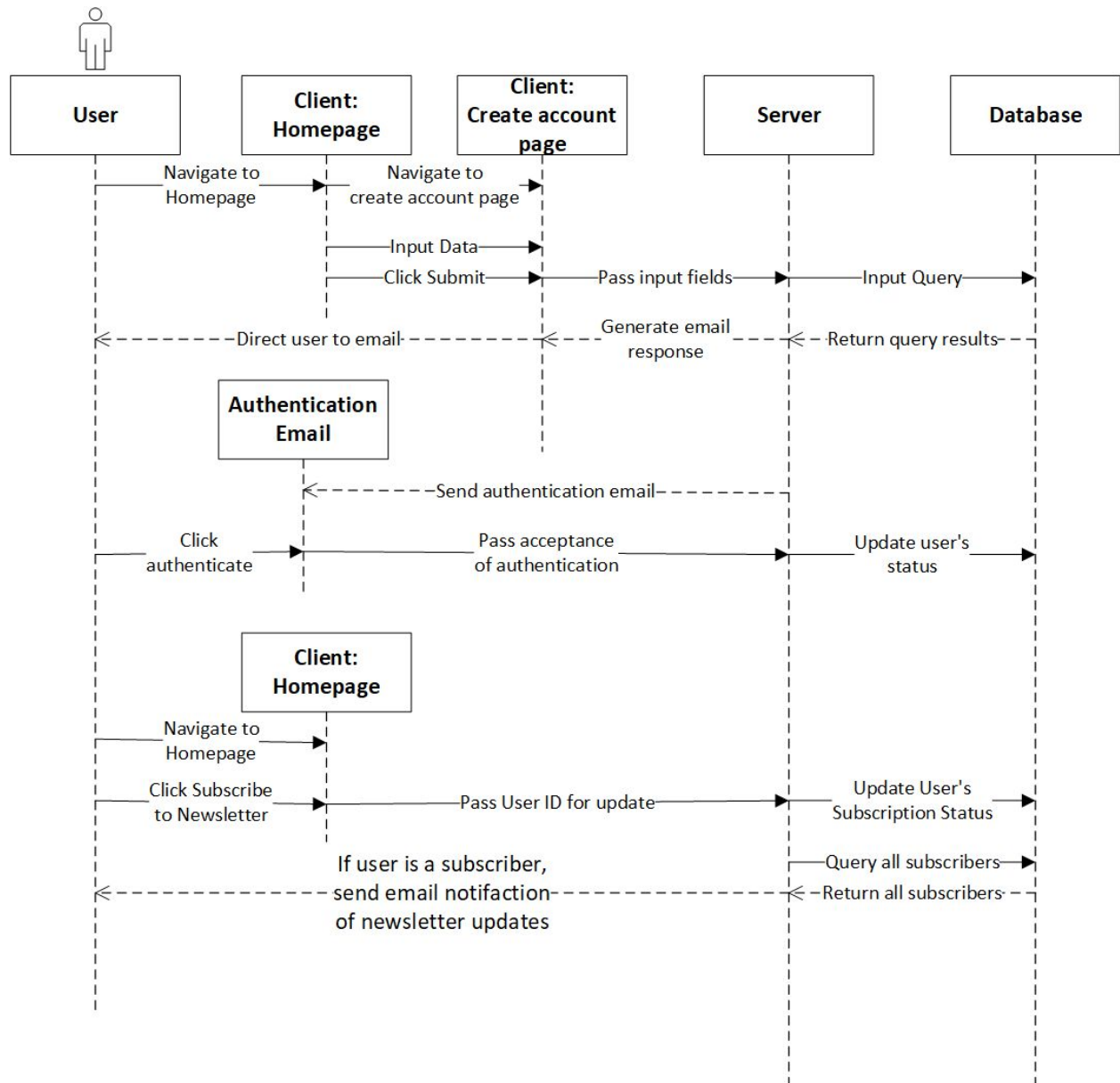
service provided, as well as redirects to the payment/download page for the service, if necessary.

- a. **Est. Implementation Time:** 14 Pair-Programming Hours
- 8. Develop a page for each user in the system for Administrator-level users, with clickable buttons that allow admins to revoke privileges for “lower-level” users. Admin-tier users should be able to click usernames in service reviews, with these clickables redirecting to the user page mentioned above.
  - a. **Est. Implementation Time:** 7 Pair-Programming Hours
- 9. “Listing” pages should automatically populate with relevant database entries based on rating. Rating should increase or decrease the position of the guide/service in the system in order to maintain proper ordering.
  - a. **Est. Implementation Time:** 2 Pair-Programming Hours
- 10. Each page should include a Go Back button that returns the user to the previous page, or even a button that returns the user all the way to the home page. Otherwise several pages, such as the user account page, will need to include redirects to the Search Guides or Search Services pages.
  - a. **Est. Implementation Time:** 1 Pair-Programming Hours
- 11. Develop clickables for the newsletter that redirect to an email entry form and a submit button. If we want to require the user to have an account in order to subscribe to the newsletter, then make this a clickable in the user account page.
  - a. **Est. Implementation Time:** 2 Pair-Programming Hours
- 12. Develop a bookmarking system that copies/adds an additional redirect to bookmarked services and guides to the user account page.
  - a. **Est. Implementation Time:** 5 Pair-Programming Hours
- 13. Develop separate permission tiers for different account levels, set when the user creates an account. If possible, add a way for the user to “upgrade” their account at a later date without creating a new account. Different permission tiers will allow certain clickables to be visible on certain pages, or add new options to the user account page.
  - a. **Est. Implementation Time:** 11 Pair-Programming Hours



## UML Sequence Diagrams and Spikes for User Stories Implementation

Create account and subscribe to a newsletter UML Sequence:



### Rate and Review Services Spike:

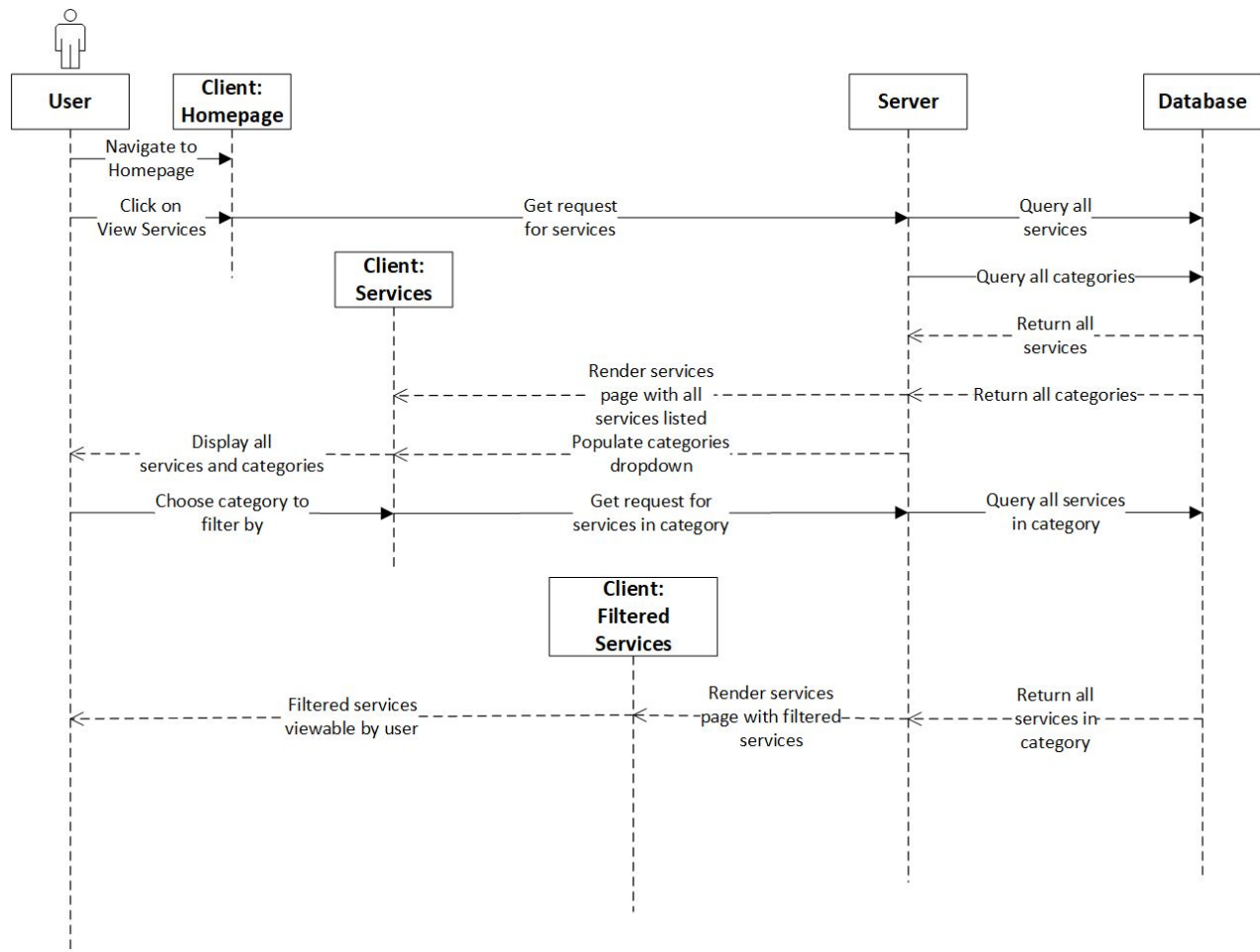
For this spike, we performed a few pieces of the implementation in isolation. Here were the steps, and an estimate of how long it would take a pair of programmers to implement the full functionality:

1. Create a database: Our db was limited, only including userID and credentialStatus for the `user` table, serviceID for the `service` table, and id, serviceID, and serviceRating for the `review` table. This took about ten minutes, so creating the full database with all the correct entities and relationships for this particular functionality will take 30 minutes to an hour. The most complicated entity would be the averageRating for the service. This is generated and updated continually by taking the average score of all ratings attached to the service.
2. Created two simple client pages: These were only created in html, but they will need to be done more completely in a .handlebars file to accept data rendering from the server. The first web page was just a formatted page that would display rows for each service, with a brief description, a button to view the service, and a button to review the service. The review service button has a hyperlink to the other page we created, which was a simple page with a dropdown that selects the rating (values 1 - 100). This only took about 15 minutes to accomplish with simple HTML, but using handlebars will take at most an hour.
3. The only piece we did not try to implement yet was the actual server. The server will take a little more work than either of the other individual steps, because we will need to create sessions for users with sessions dependent on their credentialStatus. The steps involved in this implementation will be as follows:
  - a. When user logs in, their login information will be queried to the db from the server. If the username matches a username in the db, and their password matches the password for that user, then the server will create a session.
  - b. The session id created by the server will have different values depending on the user's credentialStatus. If they have a credentialStatus that allows them to write reviews, then a handlebars page will be rendered that makes the "review service" button available. Different client pages are rendered depending on the session id, with some session ids being reserved for content authors and others reserved for regular users.
  - c. When the user creates a review, all of the information from the review will be passed to the server, then queried to the database. The server will also update the averageRating, since there is now one more review, so the averageRating must be recalculated.

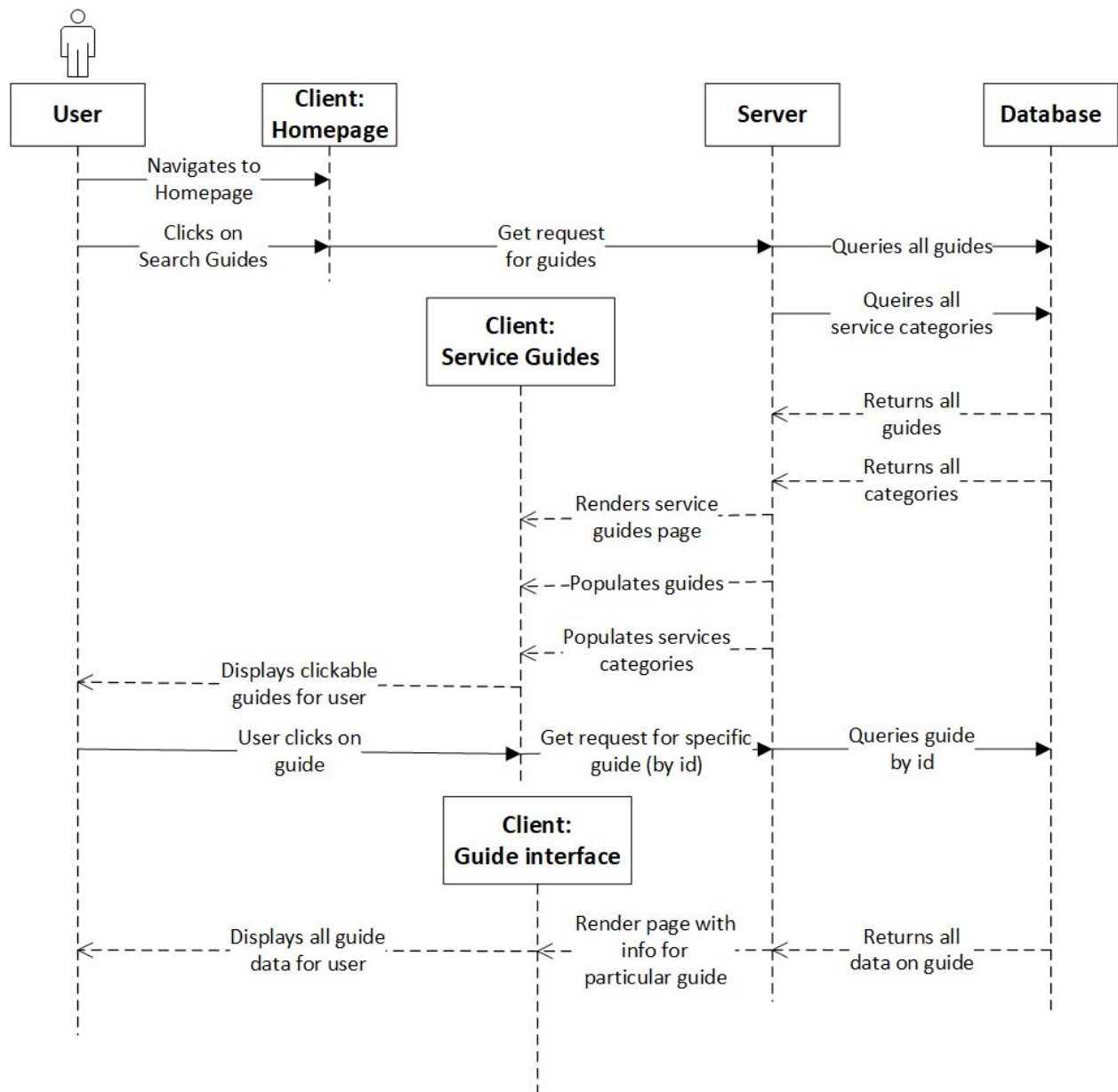
Setting up these queries and the backend functionality will take 3-4 hours for a pair of programmers.

Overall, we estimate that this implementation will take, in the worst case, 6-7 hours.

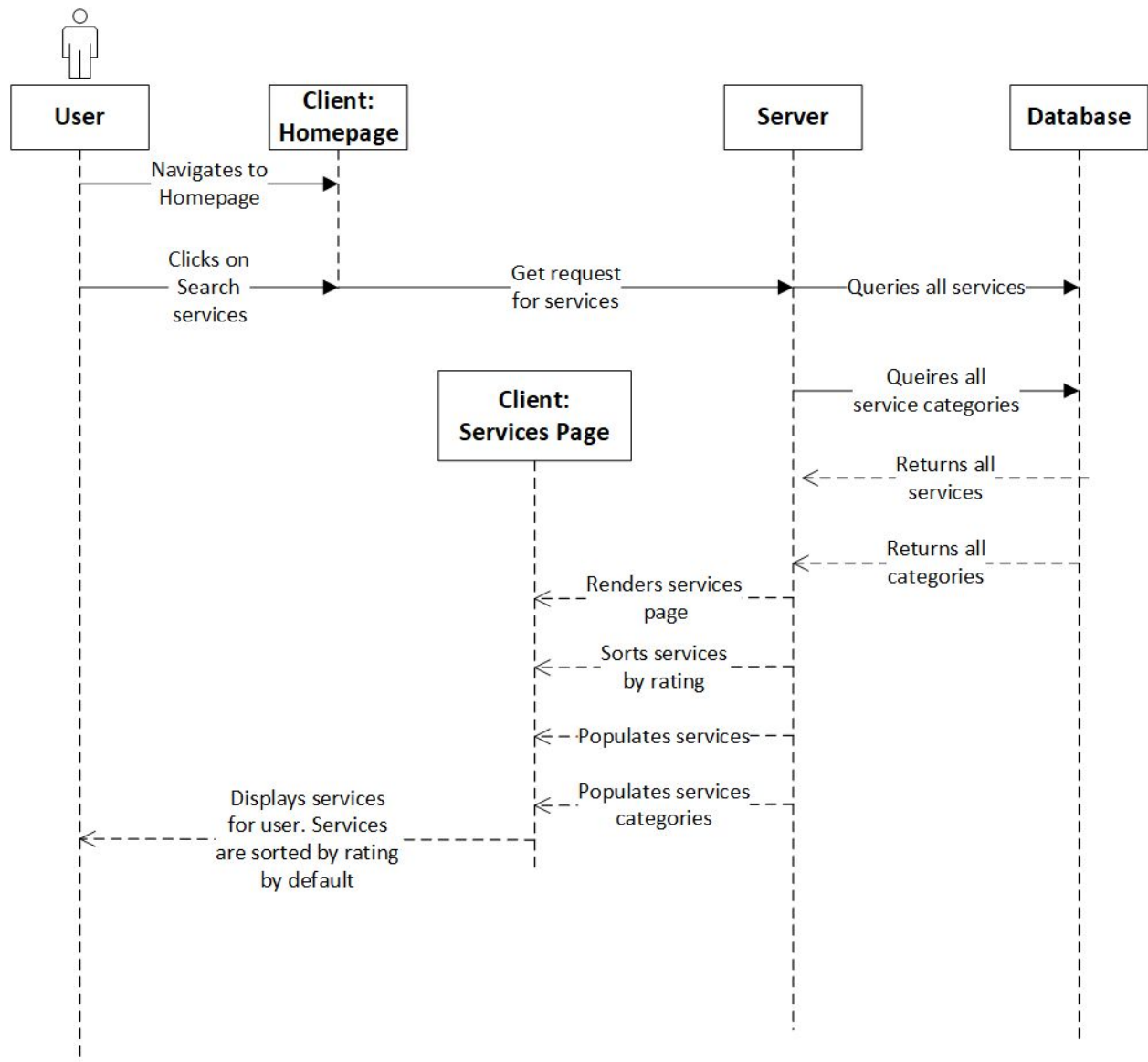
### Viewing Guide in Specific Category UML Sequence:



## Parents Protecting their Kids UML Sequence:



View top-ranked services UML Sequence:



### Manage Bookmarks Spike:

Every element of this implementation is relatively simple. We created a small database to test this out, which just had three tables: users, services, and userBookmarks. The userBookmarks only has two entities: a foreign key to users and a foreign key to services. Services has an id and a description, and users has an id. On the backend, we setup the node environment to create sessions, which are associated with the user's id. When a user clicks on the "bookmark this service" button, a small post request is sent to the server, which then inserts to the database a simple query, "INSERT INTO userBookmarks (userID, serviceID) VALUES (?, ?)", with the userID and serviceID replaced by the value stored in the sessionID (which is the user's id) and the value stored as the ID of the service, respectively. The last step of this will be to populate the user's account page with all of their bookmarks, which is exactly the same as populating the regular services page, except with the filter so that only services showing in the userBookmarks table show for that user. The queries and backend scripting took about an hour and a half, so the full implementation can be done in 4 - 5 hours.

### Work Plan Summary

- Create account and subscribe to newsletter for updates (Due Date: 5/29)
  - **Tasks:** Develop a homepage UI, a redirect button for the create account page, a form for personal info along with a button to submit the info, and a username/password validation for future logins. Also develop an account homepage, and a clickable option to subscribe to the website's newsletter from the account homepage.
  - **Est. Implementation Time:** 3 Pair-Programming Hours
  - **Programmer(s):** Sheng, Connor
- Rate and Review Service (Due Date: 5/29)
  - **Tasks:** Develop a clickable 1-5 star rating button(s) for each service page UI. Also add a button to each page for the user to add a review. The button should check to make sure the user has an account and is logged in. If the user is not logged in, clicking this button will instead redirect to the login page. If the user has the proper credentials, then clicking the Add Review button will open a form to allow the user to enter text and submit it, which will then add the review to the list of reviews on the service page.
  - **Est. Implementation Time:** 2 Pair-Programming Hours
  - **Programmer(s):** Michael, David
- Viewing Guide In Specific Category (Due Date: 6/5)
  - **Tasks:** Develop a homepage UI, and a clickable button leading to the guides page UI. From here, users can see a list of different guides in a populated list, with tabs at the top for category, date uploaded, rating, etc. Clicking on category will sort the guides by content type. Clicking on a guide name will list the title, author, date written, and a text block with information about the topic. At the

bottom of the guide page, a small list will be populated with the most popular services for that content type.

- **Est. Implementation Time:** 10 Pair-Programming Hours
- **Programmer(s):** Sheng, Connor
- Parents Protecting Their Kids (Due Date: 6/5)
  - **Tasks:** Develop homepage UI and clickable button that redirects to the Search Guides page. This page should automatically populate with the guides in the database, sorted by rating. Each guide name in the list should be a button that allows for a redirect to the page for that specific guide. The guide pages should provide a text output for the service type.
  - **Est. Implementation Time:** 10 Pair-Programming Hours
  - **Programmer(s):** Michael, David
- View top-ranked Services (Due Date: 6/5)
  - **Tasks:** Develop homepage UI and clickable services button. Clicking the services button will present a populated list of services sorted by rating. The top of the services list will have several clickable filter options, such as category for sorting services by specific technology types.
  - **Est. Implementation Time:** 9 Pair-Programming Hours
  - **Programmer(s):** Michael, Sheng
- Manage Bookmarks (Due Date: 6/5)
  - **Tasks:** Develop a user account homepage with a list of previously bookmarked guides and services based on the user's account identity. This page will have clickable delete buttons for the bookmarks, which will allow users to remove the bookmark from their user account database of bookmarked items. Develop a clickable bookmark button for each guide and service entry in the primary database that will link the viewed item to the user account identity, and copy/link the viewed item to the user's bookmark database.
  - **Est. Implementation Time:** 4 Pair-Programming Hours
  - **Programmer(s):** David, Connor

### Customer Involvement

Our customer was able to meet with us on Thursday night to come up and approve user stories, but was not able to participate on any other days.

### Group Contributions

Sheng: Wrote five user stories, add new use case flow of actions and message sequence diagram, updated class diagram in HW4.

David: Wrote five user stories, finalized effort estimation for user stories and tasks, acted as customer to set due dates, curated and finalized user stories, and message sequence diagram for Use Case #3.

Connor: Wrote five user stories, designed and finalized UML diagrams and led spike efforts.

Michael: Wrote five user stories, broke down the user stories into tasks, and summarized the work plan.